

①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ **Offenlegungsschrift**
⑩ **DE 100 58 443 A 1**

⑤1 Int. Cl.⁷:
H 04 L 12/56

②1 Aktenzeichen: 100 58 443.8
②2 Anmeldetag: 24. 11. 2000
④3 Offenlegungstag: 4. 10. 2001

DE 100 58 443 A 1

③0 Unionspriorität:
89105205 22. 03. 2000 TW

⑦1 Anmelder:
Industrial Technology Research Institute, Hsinchu
Hsien, TW

⑦3 Vertreter:
Viering, Jentschura & Partner, 80538 München

⑦2 Erfinder:
Lu, Kuo-Cheng, Hsinchu, TW; Zhao, Shi-Ming,
Teipei, TW; Yuan, Kuo-Hua, Kanhsiung, TW

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤4 Verfahren zum Klassifizieren von Datenpaketen

⑤7 Es wird ein flexibles und hochleistungsfähiges Verfahren zum Klassifizieren von Datenpaketen offenbart. Das Verfahren weist das Umwandeln der Original-Regeldatenbank in ein Regelabbildungstabellen-Format zum Speichern auf. Das Verfahren zum Erzeugen der Regelabbildungstabelle weist das Aufteilen eines Eingangsschlüssels in eine Mehrzahl von Unterschlüsseln, und das nacheinanderfolgende Vergleichen verschiedener Gruppierungskombinationen von jedem Unterschlüssel mit dem korrespondierenden Unterschlüsselfeld jeder Regel auf. Schließlich werden die Ergebnisse in der Regelabbildungstabelle unter Benutzen einer "bit map"-Methode, das heißt in Einzelbit-Darstellung, gespeichert. Die Erfindung schafft ein Verfahren zum Klassifizieren von Datenpaketen, das eine Mehrzahl von Regeldatenbanken oder Untertabellen unterstützt, so dass das gleichzeitige Vorhandensein einer Mehrzahl von Regeldatenbanken, die alle eine verschiedene Länge und Breite aufweisen können, in derselben Suchmaschine möglich ist. Zusätzlich stellt die Gestaltung tatsächliche Verbesserungen (höhere Geschwindigkeit, geringerer Platzbedarf) und Flexibilität (mögliches gleichzeitiges Vorhandensein verschiedener Regeldatenbanken) bereit. Ferner kann das Suchverfahren als Allzweck-Suchmaschine in der Form eines Netzwerkprozessors benutzt werden, oder in einer beliebigen Situation, in der eine schnelle Suche erforderlich ist. Das Suchverfahren kann auch als Ersatztechnologie für Assoziativspeicher dienen.

DE 100 58 443 A 1

Beschreibung

[0001] Die Erfindung schafft ein Verfahren zur Suche in einer Datenbank mit Platzhalter-Feldern (nicht relevante Datenabschnitte). Insbesondere schafft die Erfindung ein flexibles und hochleistungsfähiges Verfahren zum Klassifizieren von Datenpaketen, mit dem es möglich ist, Datenpakete in Untertabellen aufzuteilen, wobei die Untertabellen verschiedene Datenfeld-Breiten und Datenfeld-Tiefen aufweisen. Das Verfahren ist insbesondere nützlich für eine Pfad-Tabellensuche für Internet-Router und zum Klassifizieren von Datenpaketen oder kann auch als allgemeine Suchmaschine für einen Netzwerkprozessor dienen.

[0002] Um einen flexibleren Dienst bereitzustellen, führt ein Router nicht mehr bloß eine einfache Suche in einer Routing-Tabelle durch, und anschließend ein Umleiten eines eingehenden Datenpaketes an die nächste Workstation. Ein derzeitiger Internet-Switch/Internet-Router muss die Fähigkeiten besitzen, Datenpakete klassifizieren zu können, verschiedene Dienstgüte-Garantien (QoS) bereitzustellen oder Datenverarbeitung auf verschiedenen Sicherheitsstufen innerhalb eines virtuellen Privatnetzes ("virtual private network") durchführen zu können. Auch eine Firewall, die Netzwerksicherheit gewährleistet, beruht auf Techniken der Datenpaket-Klassifikation, um die Genehmigung zu erteilen, Zugang zu einem Netzwerk zu erhalten oder dieses zu verlassen. Anders ausgedrückt: viele neuartige Netzwerkdienste hängen sehr stark von der Klassifikation von Datenpaketen ab.

[0003] Um eine hochwertige Datenpaketklassifikation zu erreichen, ist die Fähigkeit zum Analysieren des Anfangsblocks eines Datenpaketes (Datenpaket-Header) sehr wichtig. Benutzt man beispielsweise den TCP/IP Standard, und entscheidet man sich dafür, den Anwendungsfluss zu benutzen, muss der 104-Bit-Anfangsblock, der eine IP Absender-Adresse (32 Bits), eine IP-Ziel-Adresse (32 Bits), ein Protokoll (8 Bits), eine Absender-Port-Nummer (16 Bits), und eine Ziel-Port-Nummer (16 Bits) aufweist, auf die Regel-Datenbank Bezug nehmen, um zu bestimmen, wie ein Datenpaket zu verarbeiten ist. Im Allgemeinen würde der Inhalt, den die meisten Regel-Datenbanken aufweisen, dem Netzwerkadministrator erlauben, flexible Regeln für sogenannte Platzhalter-Felder (nicht relevante Datenabschnitte) aufzustellen.

[0004] Tabelle 1 listet beispielsweise einige typische Regeln auf (Platzhalter-Felder sind mit "X" bezeichnet).

Tabelle 1

Eine typische Datenpaket-Klassifikationstabelle

Absender IP	Ziel IP	Protokoll	Absender-Port	Ziel-Port	Dienstgüte
140.96.115.X	X.X.X.X	06 (TCP)	80 (HTTP)	X	Hoch
140.96.114.X	140.96.116.X	X	X	X	Mittel
X.X.X.X	X.X.X.X	X	X	X	Gering

[0005] Aufgrund der zunehmenden Bedeutung von Datenpaket-Klassifikation wurde eine Anzahl von Artikeln betreffend Suchalgorithmen in internationalen Zeitschriften veröffentlicht. Beispielsweise haben V. Srinivasan et al. (V. Srinivasan, G. Varghese, S. Suri und M. Waldvogel "Fast and Scalable Layer 4 Switching," ACM SIGCOMM'98, Vancouver, British Columbia) ein Kreuz-Produkt- ("cross-product") Suchverfahren vorgeschlagen. T. V. Lakshman et al. (T. V. Lakshman und D. Stiliadis "High-Speed Policy-Based Packet Forwarding Using Efficient Multidimension-Range Matching" ACM SIGCOMM'98, Vancouver, British Columbia) haben vorgeschlagen, fünf Speicherbanken zum Suchen nach 1024 Regeln zu benutzen. N. McKeown et al. (N. McKeown "Packet Classification an Multiple Fields," Inforcomm 2000) hat ein Kompressions-Verfahren, basierend auf der Beobachtung von Regeleigenschaften, vorgeschlagen. Alle diese herkömmlichen Verfahren sind jedoch wenig effizient und benutzen umfangreiche Speichermengen in "worst-case"-Szenarien (Szenarien, die den schlimmsten Fall voraussetzen). Darüber hinaus sind die herkömmlichen Verfahren ungeeignet für andere Such-Typen (wie IP Pfad-Tabellen). Assoziativspeicher (content addressable memory, CAM) (T. Pei und C. Zukowaki "Put Routing Table in Silicon," IEEE Network Magazine, Seiten 42-50, Januar 1992) ist eine weitere der Techniken zum Lösen von Datenpaket-Klassifikations-Problemen. Dennoch besteht der größte Nachteil in der Benutzung von Assoziativspeichern darin, dass ein solcher Speicher gegenwärtig ziemlich teuer ist. Darüber hinaus muss eine spezielle Schaltungsgestaltung und Layout-Technik benutzt werden, falls Assoziativspeicher benutzt werden.

Folglich wurden Assoziativspeicher bis jetzt noch nicht in größerem Umfang übernommen.

[0006] Durch die Erfindung wird das Problem gelöst, ein flexibles und hochleistungsfähiges Verfahren zum Klassifizieren von Datenpaketen bereitzustellen. Das Problem wird gelöst, indem ein Verfahren zum Klassifizieren von Datenpaketen bereitgestellt wird, das die Umwandlung einer Original-Regel-Datenbank zum Speichern in das Format einer Regelabbildungstabelle bewerkstelligt. Die Regelabbildungstabelle wird dadurch gebildet, dass ein Eingangsschlüssel in eine Mehrzahl von Unterschlüssel aufgeteilt wird und dann nacheinander der Inhalt jedes Unterschlüssels mit dem Unterschlüsselfeld jeder Regel verglichen wird. Schließlich werden die Ergebnisse des Vergleichs ("1" zeigt eine Übereinstimmung an, wohingegen "0" eine Diskrepanz anzeigt) in der Regelabbildungstabelle in Einzelbit-Darstellung gespeichert.

[0007] Wenn erfindungsgemäß der Eingangsschlüssel eine Größe von W Bits aufweist, jeder Unterschlüssel G Bits aufweist und die Regeldatenbank N Regeln aufweist, so weist die Regelabbildungstabelle eine Größe (S) auf, die durch die Formel $S = (W/G) \times N \times 2^G$ (Bits) gegeben ist, und die Mindestmenge von Speicher, die bei jeder Suche ausgelesen wird (A) ist durch die Formel $A = (W/G) \times N$ (Bits) gegeben.

[0008] Wenn außerdem der Gruppierungszustand von jedem Unterschlüssel zwei ist, das bedeutet, dass sie eine Größe

von jeweils zwei Bits aufweisen ($G = 2$), wird die kleinste Regelabbildungstabelle erhalten. Die Größe der kleinsten Regelabbildungstabelle ist durch die Formel $S = (W \times N) \times 2$ (Bits) gegeben, und die zugehörige kleinste ausgelesene Speichermenge ist durch die Formel $A = (W \times N)/2$ (Bits) gegeben.

[0009] Das Verfahren zum Durchsuchen der Regelabbildungstabelle weist auf: Extrahieren jedes Unterschlüssels aus dem Eingangsschlüssel; Auslesen entsprechender Regelvektoren in der Regelabbildungstabelle, wobei die Unterschlüsselwerte direkt als Indizes verwendet werden; und Ausführen einer UND-Operation auf die Regelvektoren. Der aus der Berechnung resultierende Vektor ist als Übereinstimmungs-Regelvektor bekannt. Falls die übereinstimmende Regelabbildung von Null verschieden ist, wird das sich am weitesten links befindliche Bit (unter der Annahme, dass die am weitesten links befindliche Regel die höchste Priorität aufweist) extrahiert und stellt das Suchergebnis dar. Das Suchergebnis wird mit der Größe der zugehörigen Daten multipliziert. Gemeinsam mit der Startadresse zum Aufbewahren der zugehörigen Daten kann eine Datenspeicheradresse entsprechend dem Suchergebnis gefunden werden.

[0010] Die Erfindung benutzt eine Mehrzahl von Suchmaschinen, die alle parallel arbeiten, um die Suchoperationen in der Regelabbildungstabelle auszuführen. Jede Suchmaschine bearbeitet einen Abschnitt des Regelvektors. Indessen wird die Zuordnung der Unterschlüsselbereiche zu den Suchmaschinen durch eine Verschachtelungsmatrix erreicht.

[0011] Ferner kann die Regelabbildungstabelle in eine Mehrzahl von Untertabellen zerlegt werden, so dass die Zahl der Regeln und die Größe einer Regel in jeder Untertabelle festgesetzt werden kann. Jede Untertabelle weist ein Anfangs-Suchwertregister, ein End-Suchwertregister und ein Register zum Aufzeichnen der Größe der Untertabelle auf. Jede Untertabelle kann weiter ein Register zum Speichern der Anfangs-Speicheradresse zum Aufbewahren der zugehörigen Daten und ein Register zum Speichern der Größe des Speicherortes der zugehörigen Daten aufweisen.

[0012] Die Erfindung stellt auch ein Verfahren zur Klassifikation von Datenpaketen bereit, das eine Mehrzahl von Regeldatenbanken oder Untertabellen unterstützt. Zusätzlich erlaubt die Erfindung das gleichzeitige Vorhandensein einer Mehrzahl von Regeldatenbanken, die jeweils eine unterschiedliche Länge und Breite aufweisen können, in derselben Suchmaschine. Deshalb stellt die Ausgestaltung tatsächliche Verbesserungen (höhere Geschwindigkeit, geringerer Platzbedarf) und Flexibilität (mögliches gleichzeitiges Vorhandensein verschiedener Regeldatenbanken) bereit. Ferner kann die Erfindung nicht nur ein dynamisches Festsetzen verschiedener Größen einer Regel für Untertabellen in physikalischen Speichereinheiten bereitstellen, sondern sie kann dem Suchverfahren auch eine unbegrenzte Flexibilität bereitstellen. Kurz gesagt kann das erfindungsgemäße Suchverfahren als Allzweck-Suchmaschine in der Ausgestaltung in einem Netzwerkprozessor benutzt werden, oder in jeder beliebigen Situation, in der eine schnelle Suche nötig ist. Das Suchverfahren kann auch als Ersatztechnologie für Assoziativspeicher eingesetzt werden.

[0013] Selbstverständlich sind sowohl die vorangehende allgemeine Beschreibung als auch die folgende detaillierte Beschreibung exemplarisch und beabsichtigen, eine weitergehende Beschreibung der beanspruchten Erfindung bereitzustellen.

[0014] Ausführungsbeispiele der Erfindung sind in den Figuren dargestellt und werden im Weiteren näher erläutert.

[0015] Es zeigen:

[0016] Fig. 1 ein Diagramm, das eine Datenpaket-Klassifikations-Datenbank zeigt, die fünf 8-Bit Regeln aufweist,

[0017] Fig. 2 ein Diagramm, welches das Verfahren zum Klassifizieren von Datenpaketen gemäß einem Ausführungsbeispiel der Erfindung zeigt.

[0018] Fig. 3 ein Diagramm, das eine Regeltabelle zeigt, die eine Größe von 16 Bit und eine Eintrags-Tiefe von 32 Bit aufweist.

[0019] Fig. 4 ein Diagramm, das die Suchmaschinen in einem parallel arbeitenden System zeigt.

[0020] Fig. 5 ein Diagramm, das den Inhalt innerhalb der vier Suchmaschinen, die in Fig. 4 gezeigt sind, zeigt, und

[0021] Fig. 6 ein Ablaufdiagramm, welches das gesamte erfindungsgemäße Tabellen-Such-Verfahren gemäß einem Ausführungsbeispiel der Erfindung zeigt.

[0022] Im Weiteren wird detailliert auf bevorzugte Ausführungsformen der Erfindung Bezug genommen, von denen Beispiele in den zugehörigen Figuren veranschaulicht sind. Wo immer möglich, werden die selben Bezugsziffern in den Zeichnungen und der Beschreibung benutzt, um identische oder entsprechende Teile zu bezeichnen.

[0023] Durch die Erfindung wird ein sehr schnelles und ökonomisches Suchverfahren geschaffen, mit dem insbesondere Daten durchsucht werden können, die Platzhalter-Bitfelder, das heißt Datenabschnitte mit nicht relevantem Inhalt, aufweisen. Um die Erklärung zu vereinfachen, wird für die folgende Veranschaulichung eine Regeltabelle ausgewählt, die nur fünf Regeln aufweist. Fig. 1 ist ein Diagramm, das eine Datenpaket-Klassifikations-Datenbank zeigt, die fünf 8-Bit-Regeln aufweist. Wie in Fig. 1 gezeigt, weist die Regeltabelle 100 fünf 8-Bit-Regeln auf, wobei jedes der Regel-Bits eine "1", eine "0" oder ein "X" (Platzhalter) aufweist. Das erfindungsgemäße Suchverfahren ist fähig, eine Regel zu finden, die mit einem bestimmten Eingangsschlüssel in der Regeltabelle 100 übereinstimmt. Wenn mehr als eine Regel mit dem Eingangsschlüssel übereinstimmt, gibt das Suchverfahren die am weitesten vorne in der Regeltabellenliste befindliche Regel (höherwertigstes Bit) als Suchergebnis aus. Benutzt man beispielsweise einen Eingangsschlüssel #0 ("11110011"), so sind die übereinstimmenden Regeln in Tabelle 100 die Regel #1 ("11110x11"), die Regel #2 ("1xxx0011") und die Regel #4 ("xxxxxx"). Unter diesen Umständen wählt das Suchverfahren Regel #1 als Suchergebnis 110 aus.

[0024] Im Folgenden wird das Verfahren zum Klassifizieren von Datenpaketen gemäß einem Ausführungsbeispiel der Erfindung beschrieben. Zunächst wird die Original-Regeltabelle in eine Regelabbildungstabelle umgewandelt und in einer Speichereinheit gespeichert. Fig. 2 ist ein Diagramm, welches das Verfahren zum Klassifizieren von Datenpaketen gemäß einem Ausführungsbeispiel der Erfindung zeigt. Die Abbildungstabelle 120, die in Fig. 2 gezeigt ist, wird durch Aufteilung des Eingangsschlüssels in eine Mehrzahl von Unterschlüssel erzeugt. Danach werden die verschiedenen Kombinationen der Bitwerte in jedem Unterschlüssel mit den entsprechenden Unterschlüsselwerten jeder Regel verglichen. Die Ergebnisse des Vergleichs werden in der Abbildungstabelle 120 entsprechend der "bit map"-Methode, das heißt in Einzelbit-Darstellung, gespeichert ("1" stellt eine Übereinstimmung dar, wohingegen "0" eine Diskrepanz darstellt). In einer Spalte #K einer Zeile der Abbildungstabelle 120 steht also genau dann eine "1", wenn der mit der Zeile korrespondierende Unterschlüsselwert mit dem durch den Unterschlüssel festgelegten Abschnitt in der Regel #K über-

einstimmt. Andernfalls steht in der Abbildungstabelle 120 an der beschriebenen Stelle eine "0". In der weiteren Beschreibung wird der Regelvektor (I, J) benutzt, um die umgewandelte Einzelbit-Darstellung der Abbildungstabelle 120 zu bezeichnen, wenn der Wert des I-ten Unterschlüssels gleich dem J-ten Unterschlüsselwert ist. Eine Zeile des Regelvektors ist also eine binäre Zahlenfolge mit einer solchen Anzahl von Binärziffern, wie die Regeltabelle Regeln aufweist. Der Regelvektor wird gebildet, indem sequentiell diejenigen Zeilen der Abbildungstabelle 120 zeilenweise angeordnet werden, für die der jeweilige Unterschlüsselwert mit dem entsprechenden Unterschlüssel übereinstimmt.

[0025] Wenn man beispielsweise die Regelabbildungstabelle 100 in Fig. 1 benutzt, und wenn man den 8-Bit Eingangsschlüssel in vier 2-Bit Unterschlüssel aufteilt, erhält man die Regelabbildungstabelle 120, die in Fig. 2 gezeigt ist. Für den Unterschlüssel #0 (Bit #1, Bit #0) des Eingangsschlüssels ist der Regelvektor (0,0), der dem Unterschlüssel #0 mit dem Unterschlüsselwert 0 entspricht, {1,0,0,1,1}. Dies zeigt, dass wenn Bit #1 und Bit #0 des Eingangsschlüssels gleich {0,0} ist, die Regel #0, die Regel #3 und die Regel #4 passen. Aus einer ähnlichen Überlegung ergibt sich, dass der Regelvektor (0,3), der dem Unterschlüssel #0 mit dem Unterschlüsselwert 3 entspricht, {0,1,1,1,1} ist. Dies zeigt, dass wenn Bit #1 und Bit #0 des Eingangsschlüssels {1,1} ist, Regel #1, Regel #2, Regel #3 und Regel #4 passen.

[0026] Durch die Errichtung der Regelabbildungstabelle 120 werden die Suchoperationen sehr stark vereinfacht. Es ist nur noch erforderlich, all diejenigen Regelvektoren zu extrahieren, die den Suchschlüsselwerten entsprechen. Benutzt man den Eingangsschlüssel #1 ("11110011") in Fig. 2 als Beispiel, werden die vier Regelvektoren Regelvektor (0,3), Regelvektor (1,0), Regelvektor (2,3) und Regelvektor (3,3) benötigt (in Fig. 2 schattiert dargestellt). Eine logische UND-Operation wird auf die Regel-Vektoren angewendet, um den Übereinstimmungs-Regelvektor zu erhalten: {0,1,1,0,1}. Dies zeigt, dass der Eingangsschlüssel mit der Regel #1, der Regel #2 und der Regel #4 übereinstimmt. Schließlich werden ein UND-Logikelement 130 und ein Prioritätskodierer 140 benutzt, um die am weitesten links befindliche Regel zu extrahieren, das heißt in diesem Fall Regel #1. Deshalb ist Regel #1 das Suchergebnis 150.

[0027] Eine klare Analyse der Leistungsfähigkeit des Suchverfahrens erfolgt auf Basis der Parameter der Auflösung (G), welche die Anzahl der Bits in jedem Unterschlüsselfeld angibt, der Größe (W), welche die Größe jeder Regel in der Regeltabelle (oder die Größe des Eingangsschlüssels) angibt, und der Anzahl (N), welche die Anzahl der Regeln in der Regeltabelle angibt, die benutzt werden (in obigem Beispiel ist $G = 2$, $W = 8$, $N = 5$). Die zum Speichern der Abbildungstabelle 120 erforderliche Speichermenge (S) ist durch die Formel gegeben:

$$S = (W/G) \times N \times 2^G \text{ (Bits)} \quad (1); \text{ und}$$

die bei jeder Suche aus dem Speicher zu lesende Speichermenge (A) ist durch die Formel gegeben:

$$A = (W/G) \times N \text{ (Bits)} \quad (2)$$

[0028] Gemäß den Werten in obigem Beispiel ($G = 2$, $W = 8$, $N = 5$) ist die erforderliche Speichermenge zum Speichern der Regelabbildungstabelle 120 gleich $(8/2) \times 5 \times 2^2 = 80$ Bits. Das aus dem Speicher auszulesende Volumen ist bei jeder Suchoperation gleich $(8/2) \times 5 = 20$, in absoluter Übereinstimmung mit dem in Fig. 2 Gezeigten. Ein besonderes Phänomen kann auch anhand der Formeln (1) und (2) beobachtet werden. Wenn $G = 1$ oder $G = 2$ ist, ist der zum Speichern der Regelabbildungstabelle erforderliche Speicherbedarf am kleinsten:

$$S = (W/1) \times N \times 2^1 = (W/2) \times N \times 2^2 = W \times N \times 2 \text{ (Bits)} \quad (3)$$

[0029] Jedoch ist für den Gruppierungszustand mit $G = 2$ die bei jeder Suche erforderliche zu lesende Speichermenge nur halb so groß wie für den Gruppierungszustand $G = 1$. Folglich produziert ein Gruppierungszustand mit $G = 2$ nicht nur einen minimalen Speicherbedarf für eine gegebene Regelabbildungstabelle, sondern erfordert auch weniger Speicherleseoperationen.

[0030] In der folgenden Beschreibung wird eine 16 Bit große Regeltabelle mit einer Eintrags-Tiefe von 32 Bit als Beispiel gewählt. Ein Gruppierungszustand $G = 2$ wird benutzt, um die Regelabbildungstabelle zu erzeugen. Wie zuvor gesagt, produziert ein Gruppierungszustand mit $G = 2$ die kleinste Speichergröße zum Speichern der Regelabbildungstabelle. Wie oben bemerkt hat diese Art von Regelabbildungstabelle insgesamt 32 Regelvektoren (ein 16 Bit Eingangsschlüssel ist äquivalent mit 8 Unterschlüsseln, und jedes Unterschlüsselfeld bildet 4 Regelvektoren ab, die jeweils 32 Bit aufweisen). Fig. 3 ist ein Diagramm, das eine Regeltabelle zeigt, die eine Größe von 32 Bit und eine Eintrags-Tiefe von 32 Bit aufweist. In Fig. 3 weist der Regelvektor [1][31 : 0] eine Einzelbit-Darstellung auf, das mit der Regel übereinstimmt, wenn das Unterschlüsselfeld #0 des Eingangsschlüssels 2'b01 ist. Ähnlich weist der Regelvektor [7][31 : 0] eine Einzelbit-Darstellung auf, die mit der Regel übereinstimmt, wenn das Unterschlüsselfeld #1 des Eingangsschlüssels 2'b11 ist.

[0031] Um die Regelabbildungstabelle zu durchsuchen, müssen die acht Regelvektoren nacheinander ausgelesen werden und der überlappende Abschnitt muss gleichzeitig extrahiert werden. Um die Suchgeschwindigkeit zu erhöhen, werden vier Suchmaschinen benutzt, die alle parallel arbeiten. Jede Suchmaschine ist für zwei Unterschlüsselfelder verantwortlich. Mit anderen Worten müssen zwei Regelvektoren (insgesamt 64 Bits) unter den acht Regelvektoren ausgelesen werden, um eine UND-Berechnung durchzuführen. Da ein weiter links befindliches Bit (höherwertiges Bit) eine höhere Priorität aufweist, wird der linkshändige Abschnitt des Regelvektors mit erster Priorität bearbeitet (der UND-Zyklus #0 ist nachfolgend beschrieben). Falls keine bekannte Regelnummer nach der Suche übereinstimmt, wird auch der rechte händige Abschnitt des Regelvektors nacheinander durchsucht (der UND-Zyklus #1 ist nachfolgend beschrieben).

[0032] Im Allgemeinen müssen physikalische Aspekte hinsichtlich der gegenwärtigen Herstellungsverfahren wie Speichergröße und Geschwindigkeit des Prioritätskodierers berücksichtigt werden. Für eine Suchmaschine mit einer 16 Bit Speichergröße sind insgesamt vier Lesezyklen erforderlich, um zwei Regelvektoren auszulesen. Da die vier Suchmaschinen parallel arbeiten, ist die gesamte Suchzeit bei der Benutzung von vier Suchmaschinen verglichen mit der gesamten Suchzeit bei der Benutzung einer Suchmaschine für die Lesezyklen dieselbe.

[0033] Fig. 4 ist ein Diagramm, das die Suchmaschinen in einem parallel arbeitenden System zeigt. Man beachte, dass die Suchmaschine #0 für die Regelvektoren bezogen auf das Unterschlüsselfeld #0 und das Unterschlüsselfeld #4 verantwortlich ist, anstatt für das Unterschlüsselfeld #0 und das Unterschlüsselfeld #1. Diese Art von Ablaufsteuerung der Regelvektoren wird oft als Verschachtelungsmatrix-Zuordnung bezeichnet. Ein Vorteil der Verschachtelungsmatrix-Zuordnung besteht darin, dass eine gleiche Anzahl von Unterschlüsselfeldern selbst dann den jeweiligen Suchmaschinen für die Parallelverarbeitung zugeordnet werden können, wenn eine kürzere Regel (beispielsweise 8 Bits) unterstützt werden muss.

[0034] Fig. 5 ist ein Diagramm, das die Inhalte innerhalb der vier Suchmaschinen, die in Fig. 4 gezeigt sind, zeigt. In diesem Beispiel sind die ersten beiden fertigen Zyklen (bekannt als UND-Zyklus #0) jeder Suchmaschine verantwortlich für die UND-Berechnung der ersten 16 Bit jedes Regelvektors. In Fig. 5 stellt $V(I,J,K)[15:0]$ den Auslesevektor jedes Lesezyklus dar, wobei I eine bestimmte Suchmaschine bezeichnet, J einen bestimmten UND-Zyklus bezeichnet und K einen bestimmten Lesezyklus bezeichnet.

[0035] Fig. 6 ist ein Ablaufdiagramm, welches das gesamte erfindungsgemäße Tabellen-Such-Verfahren zeigt. Beginnend mit dem ersten UND-Zyklus #0 werden die ersten acht Vektoren, die durch die vier Suchmaschinen ausgelesen werden, durch eine logische UND-Operation miteinander verknüpft. Nach Durchlaufen des Prioritätskodierers 200 kann festgelegt werden, ob der Eingangsschlüssel mit einer der Regeln 0 bis 15 übereinstimmt. Falls keine solche Regel nach dem UND-Zyklus #0 gefunden worden ist, wird ein nachfolgender UND-Zyklus #2 ausgeführt. Nach dem Extrahieren der jeweiligen Regelvektoren, wird eine 16-Bit UND-Berechnung durchgeführt. Falls eine geeignete Regel gefunden wird, gibt der Prioritätskodierer 210 eine Regelzahl (0 ... 15) aus. Nachdem der Wert 16 zur Regelzahl addiert worden ist (da Phase #2 die verbleibenden 16 Regeln handhabt), wird eine übereinstimmende Regelzahl erhalten.

[0036] Mittels der beschriebenen Technik wird die Suche in einer Regelabbildungstabelle erreicht, indem jede Suchmaschine einen Abschnitt des Unterschlüsselfeldes parallel verarbeitet. Durch Benutzen der Verschachtelungsmatrix ist das Suchverfahren sogar dazu fähig, Regeln mit verschiedenen Größen zu bearbeiten. Ferner kann die Erfindung eine Mehrzahl von Regeldatenbanken oder Untertabellen unterstützen. Das einzige Kriterium ist das Festsetzen des Anfangs-Suchwertes, des End-Suchwertes und der Größe der Regel der gewünschten Such-Untertabelle. Nach dem Festsetzen des Anfangs-Suchwertes kann die Suchmaschine automatisch nach der Untertabelle suchen. Folglich kann eine Mehrzahl von Regeldatenbanken, die jeweils eine unterschiedliche Länge und Breite aufweisen können, innerhalb derselben Suchmaschine gleichzeitig vorhanden sein, so dass die Betriebseigenschaften (wie Geschwindigkeit und Platzbedarf) und die Flexibilität (das gleichzeitige Vorhandensein verschiedener Regeldatenbanken) verbessert werden.

[0037] Zusätzlich hat jede Untertabelle einen Anfangsadressspeicher zum Aufbewahren der zugehörigen Daten und einen Speicher zum Aufbewahren der Größe der zugeordneten Daten. Durch Multiplizieren der Suchausgabe-Regelnummer mit der Größe der zugeordneten Daten und durch Addieren der Anfangsadresse der zugehörigen Daten im Speicher wird die Adresse für die aufbewahrten Regeldaten gefunden.

[0038] Zusammenfassend schafft die Erfindung ein flexibles und effizientes Verfahren zum Klassifizieren von Datenpaketen, das nicht nur dynamisch eine Mehrzahl von Untertabellen zuordnet, die alle eine unterschiedliche Größe einer Regel in einer physikalischen Speichereinheit aufweisen, sondern das auch die Flexibilität des Such-Verfahrens erhöhen kann. Beispielsweise kann eine 128 Bit große Untertabelle aufgeteilt werden, um die Datenfluss-Klassifikation zu spezialisieren; eine 32 Bit große Untertabelle kann aufgeteilt werden, um die Abfrage des IP-Routing-Pfads zu spezialisieren; und eine 64 Bit große Untertabelle kann aufgeteilt werden, um eine MAC Adresse unterzubringen. Ferner kann das flexible und hochleistungsfähige Suchverfahren auf die Gestaltung des Netzwerkprozessors angewendet werden, und kann so als Allzweck-Suchmaschine benutzt werden.

[0039] Die Erfindung kann in allen Sachverhalten benutzt werden, die eine schnelle Suche erfordern. Zusätzlich kann die Erfindung als Ersatztechnik für Assoziativspeicher benutzt werden.

[0040] Beispielsweise kann die Erfindung für einen LAN-Switch-Router, einen vierschichtigen Switch, ein virtuelles Privatnetz, für Messungen des Netzwerkverkehrs, für Netzwerkprozessoren, zur Datenfluss-Klassifikation, für Firewallprodukte usw. angewendet werden.

Patentansprüche

1. Verfahren zum Klassifizieren von Datenpaketen, das aufweist:
Umwandeln einer Original-Regeldatenbank zum Speichern in einem Tabellenformat einer Regelabbildungstabelle (120), wobei das Verfahren zum Erzeugen der Regelabbildungstabelle (120) aufweist:

Aufteilen eines Eingangsschlüsselwerts in eine Mehrzahl von Unterschlüsselwerten;
sequentielles Vergleichen verschiedener Kombinationen von allen Unterschlüsselwerten mit den entsprechenden Unterschlüsselfeldern jeder Regel; und
Speichern des Vergleichsergebnisses in der Regelabbildungstabelle (120) mittels des Einzelbit-Darstellungs-Verfahrens.

2. Verfahren nach Anspruch 1, bei dem die Größe (S) der Regelabbildungstabelle (120) gegeben ist durch die Formel:

$$S = (W/G) \times N \times 2^6 \text{ (Bits); und bei dem}$$

die gelesene Speichermenge (A) bei jeder Suche durch die Formel gegeben ist:

$$A = (W/G) \times N \text{ (Bits), wobei}$$

W die Größe des Eingangsschlüssels in Bits ist, G die Anzahl der Bits in jedem Unterschlüssel ist und N die Anzahl der Regeln in der Regeldatenbank ist.

3. Verfahren nach Anspruch 2, bei dem die kleinste Größe der Regelabbildungstabelle (120) durch die Formel gegeben ist:

$S = (W \times N) \times 2$ (Bits), und bei dem

die minimal gelesene Speichermenge (A) bei jeder Suche entsprechend der kleinsten Größe der Regelabbildungstabelle (120) durch die Formel gegeben ist:

$A = (W \times N)/2$ (Bits).

wenn jeder Unterschlüssel einen Gruppierungszustand mit einer Größe von 2 aufweist (das heißt $G = 2$).

4. Verfahren nach einem der Ansprüche 1 bis 3, bei dem ein Schritt bei der Suche in der Regelabbildungstabelle (120) aufweist:

Extrahieren von Unterschlüsseln aus dem Eingangsschlüssel, die als direkte Indizes dienen, um der Regelabbildungstabelle (120) korrespondierende Regelvektoren zu entnehmen; und
Durchführen einer UND-Operation auf die Regelvektoren zum Berechnen des Übereinstimmungs-Regelvektors und Extrahieren des am weitesten links befindlichen Bits, welches das Suchergebnis (150) darstellt, falls die übereinstimmende Regelabbildung von Null verschieden ist.

5. Verfahren nach Anspruch 4, bei dem die Suche in der Regelabbildungstabelle (120) unter Benutzen einer Mehrzahl von parallel arbeitenden Suchmaschinen ausgeführt wird, wobei jede Suchmaschine einen Abschnitt des Regelvektors bearbeitet, und wobei die Zuordnung der Unterschlüsselfelder zu den Suchmaschinen mittels eines Verschachtelungsmatrix-Verfahrens erfolgt.

6. Verfahren nach Anspruch 4 oder 5, bei dem das Suchergebnis (150) mit der Größe der zugehörigen Daten multipliziert wird, und zu der Anfangsspeicheradresse zum Aufbewahren der zugehörigen Daten addiert wird, wodurch eine Adresse zum Aufbewahren der Daten entsprechend dem Suchergebnis (150) gefunden wird.

7. Verfahren nach einem der Ansprüche 1 bis 6, bei dem die Regelabbildungstabelle (120) in eine Mehrzahl von Untertabellen aufgeteilt wird, wobei jede Tabelle eine festgelegte Größe einer Regel und eine festgelegte Anzahl von Regeln aufweist.

8. Verfahren nach Anspruch 7, bei dem jede Untertabelle einen Speicher zum Aufnehmen eines Anfangs-Suchwertes, einen Speicher zum Aufnehmen eines End-Suchwertes und einen Speicher zum Aufnehmen der Größe der Untertabelle aufweist.

9. Verfahren nach Anspruch 7, bei dem jede Untertabelle einen Speicher zum Aufbewahren von Informationen hinsichtlich des Speicherns der Anfangs-Speicheradresse der zugehörigen Daten und einen Speicher zum Aufbewahren von Informationen hinsichtlich des Speicherns der zugehörigen Datengröße aufweist.

Hierzu 6 Seite(n) Zeichnungen

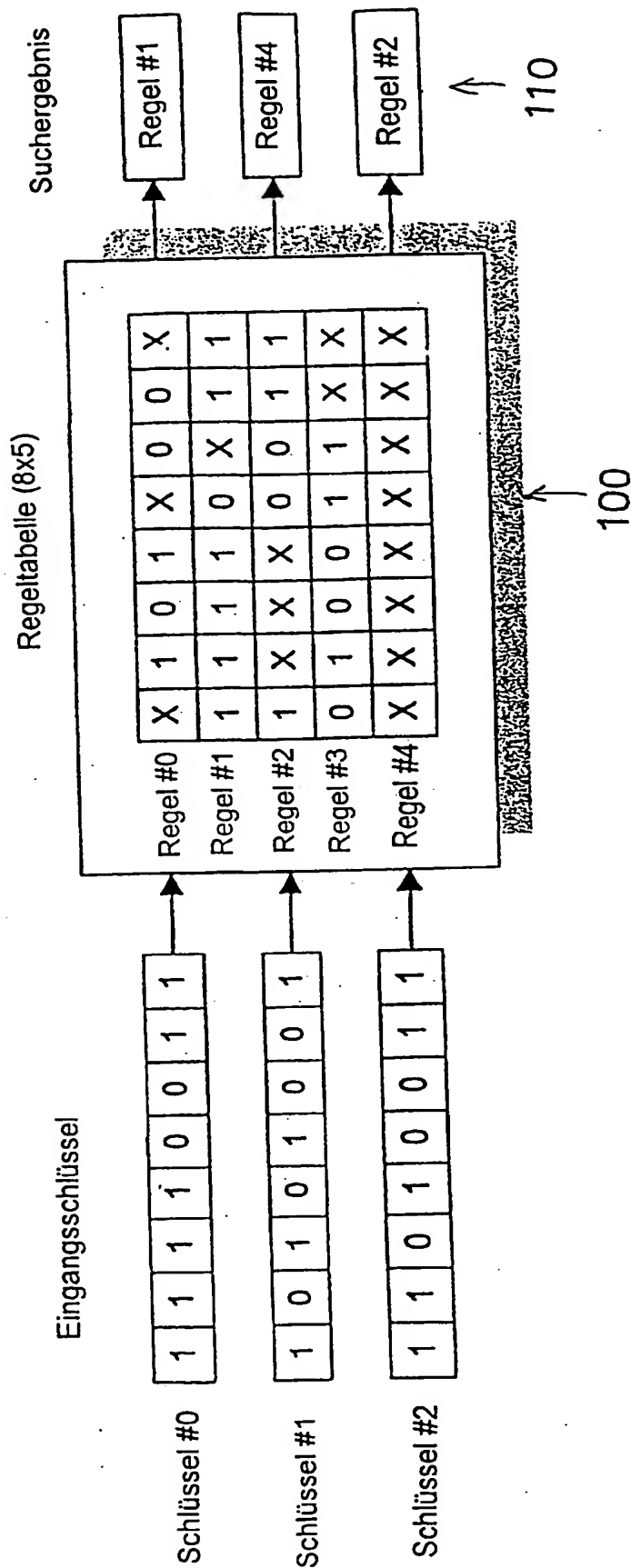


FIG. 1

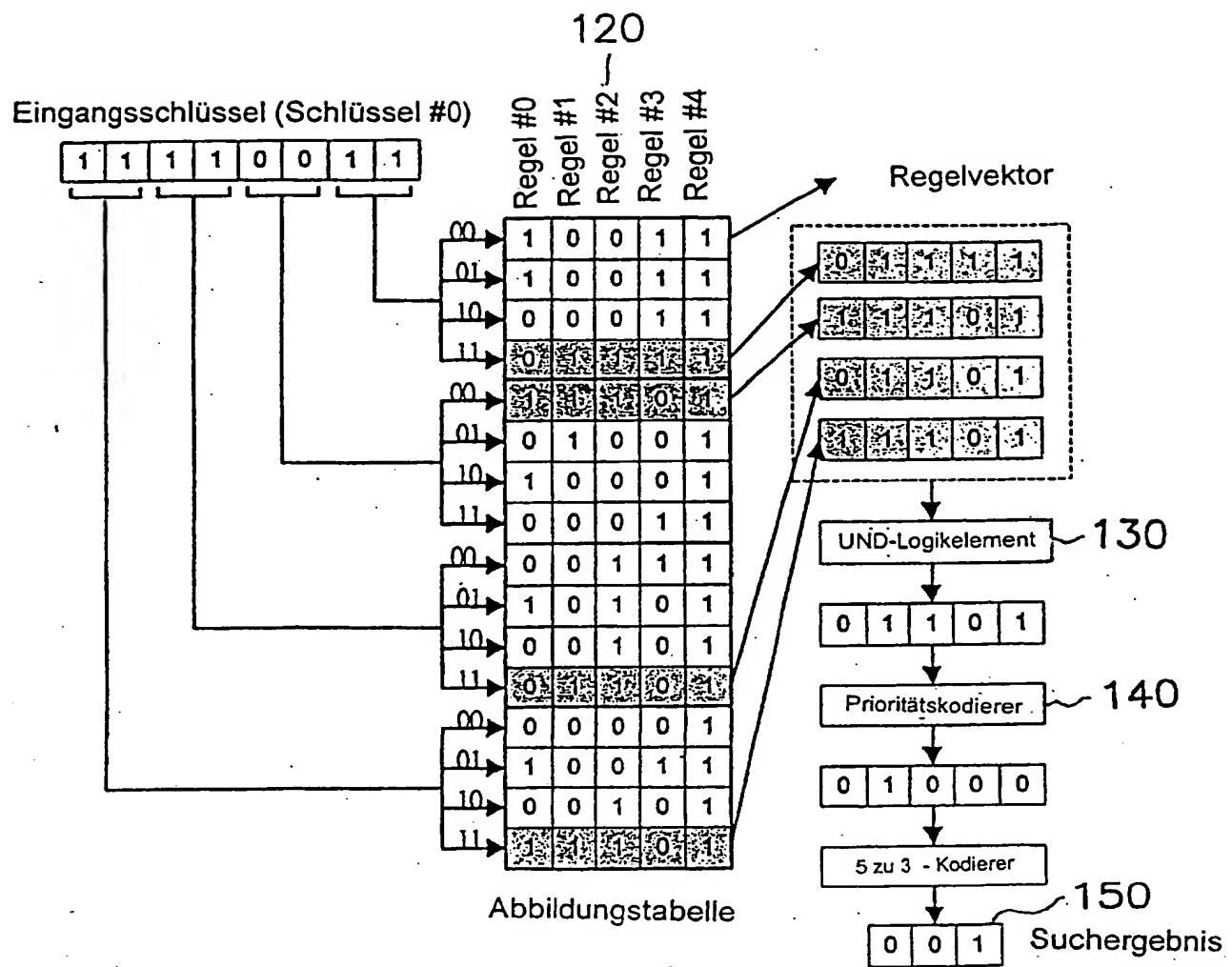


FIG. 2

Probe Eingangsschlüssel (16 bits)

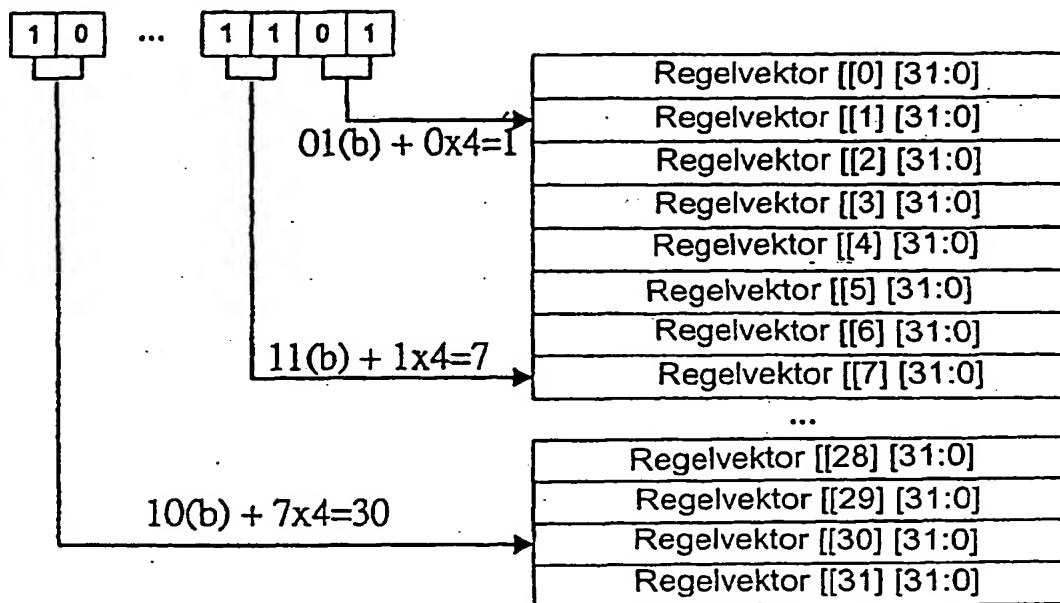


FIG. 3

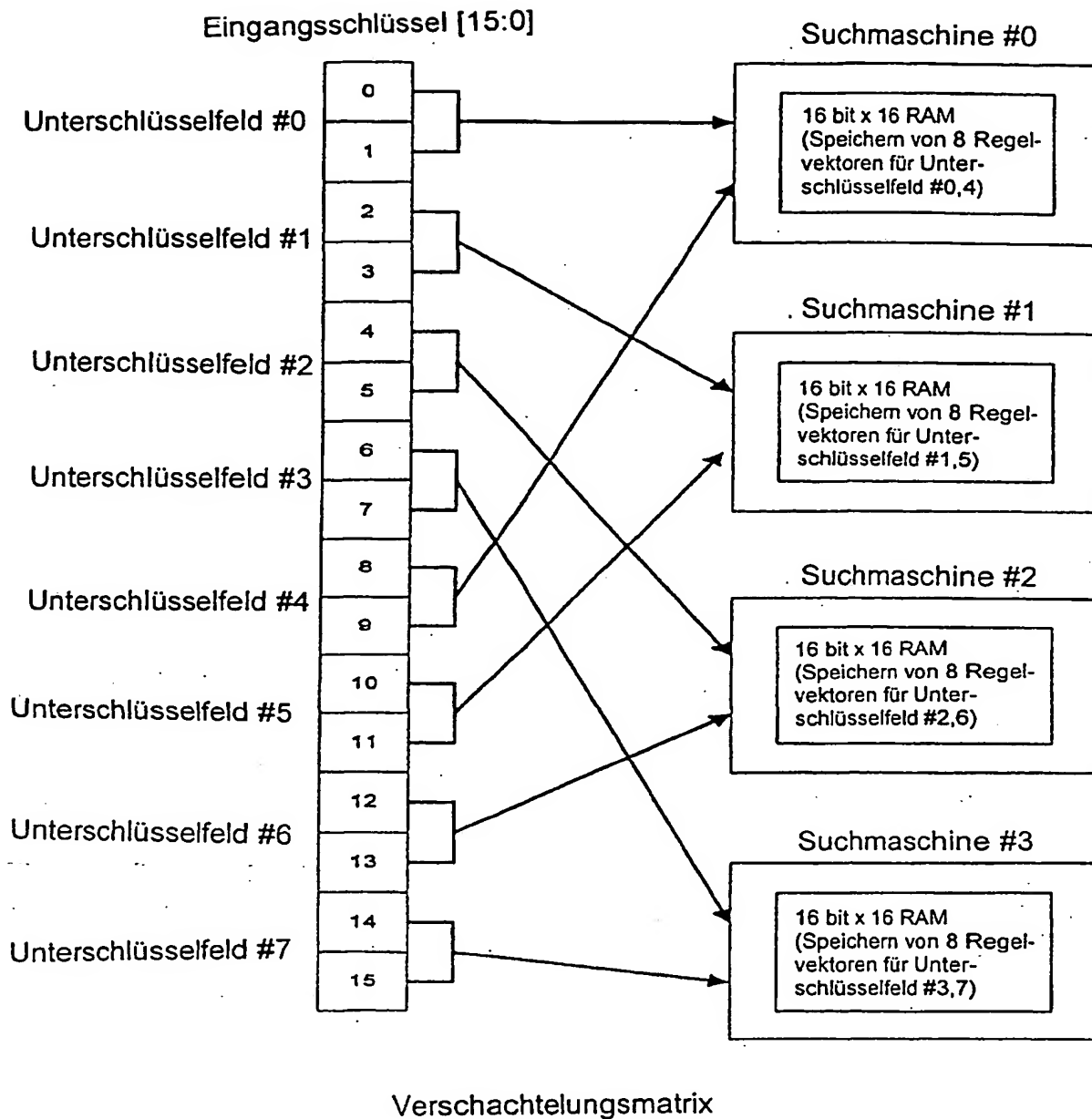


FIG. 4

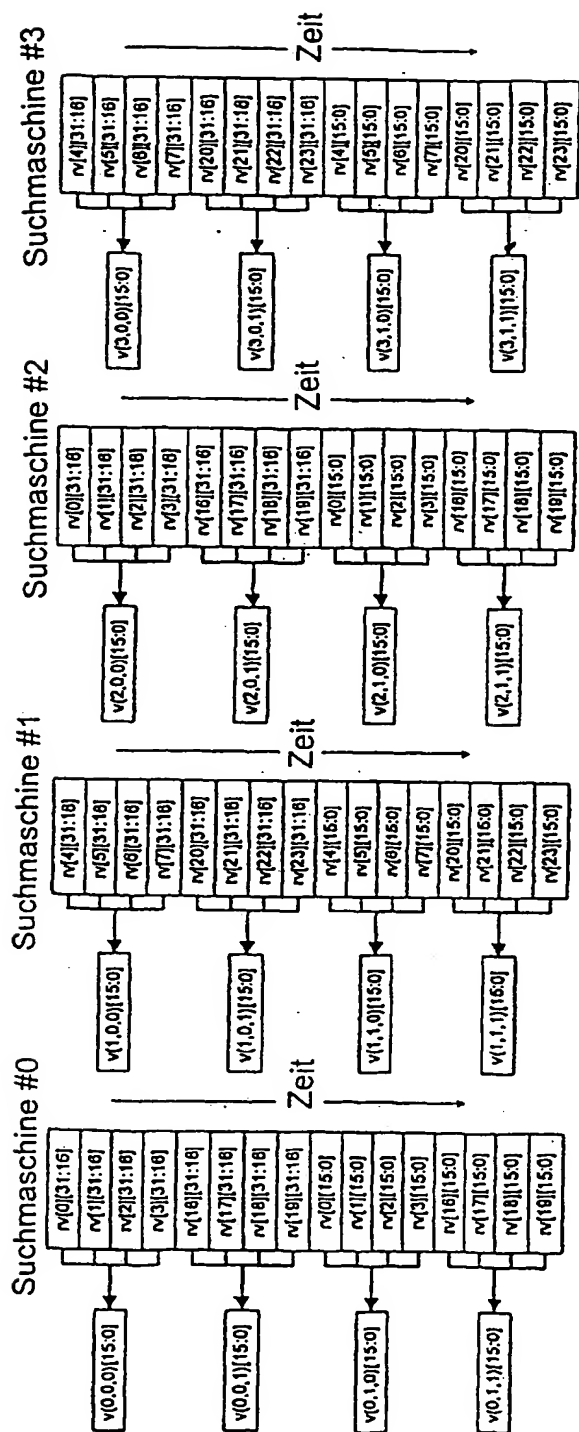


FIG. 5

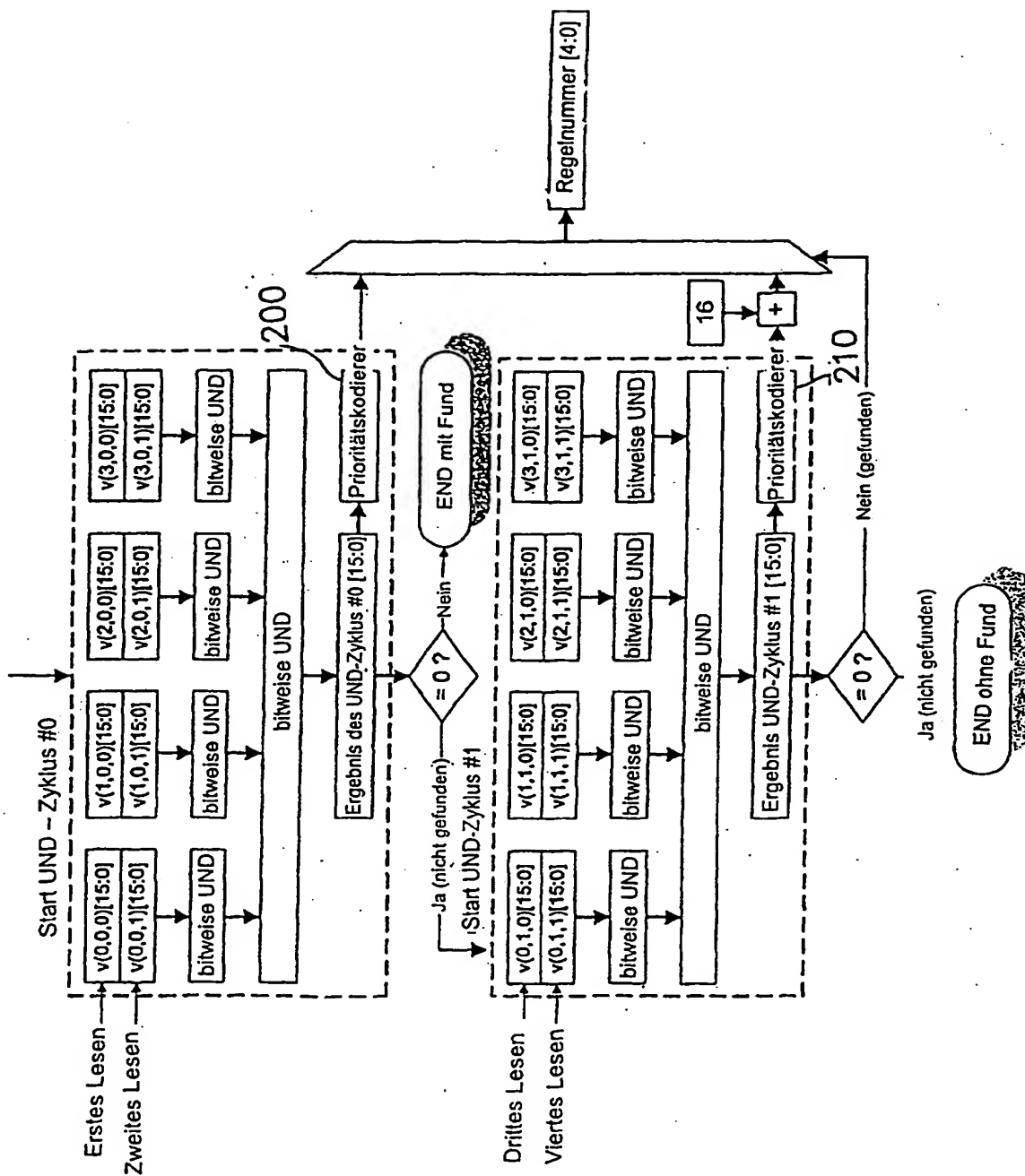


FIG. 6